Cafe Scheduling Optimization Simulation
Analysis and Validation
Repository: https://github.com/Baesiann/CS4632_Kenneth_Burke

Kenneth Burke
Department of Computer Science
Kennesaw State University
Kennesaw, Georgia 30144
Email: kburke36@students.kennesaw.edu

1. Introduction

1.1. Simulation Summary

The simulation begins with the generation of customer objects from a doubly stochastic process. Each generated customer waits in a queue, and an available barista takes the order of whoever is first in queue. The customer's order is dependent on the order distribution. Each customer also has a patience averaging 9 minutes, the customer dropping out of the queue if their wait exceeds their patience.

1.2. What is being analyzed and why

After the simulation runs, there is a tab to visualize data. The Average Wait Time per Day graph gives insight to how long customers are waiting for their drink on average, this is useful to see whether baristas are being properly allocated. Total Revenue per Day is useful to see whether the schedule manager is working properly to increase total revenue, and works as a base of comparison with other metrics. A case of this could be analysis of throughput of customers as well as customer arrivals. The Average Customers in Queue useful in the telling of how congested the system is, the higher the value the more congested it is; it is essentially a measure of how many customers are waiting in the queue at any given time. Dropped customers per day is useful for customer satisfaction and analysis of 'lost revenue,' as it is typically not a good sign if customers are losing patience and leaving the queue. The Customer Arrivals per Day gives an idea of how many people enter the coffee shop each day, and can be cross referenced with throughput and revenue to see how closely aligned those statistics are. It is inherently a good metric to show.

2. Parameter Analysis

2.1. How changing inputs affects outputs

To tackle this section, I will list the parameters that can be changed and list the changes that it makes to the simulation.

- Setup Parameters
  - Simulation Days: The amount of days the simulation will run for. The higher the amount of days, the more days the simulation will generate arrivals and metrics for.
  - Starting Baristas: The initial amount of baristas scheduled. It affects the first day of the simulation, where the rest of the days the schedule manager will decide how many baristas to schedule.
  - Seed: Can be set to have a consistent stream of random variables.
- Customer Arrival Parameters
  - Baseline Arrival Rate: This is the base arrival rate of customers, a higher rate means more customer arrivals per minute.
  - Morning Rush Intensity: This is how much more volume is expected for a morning rush, it directly affects the baseline arrival rate by creating a peak of baseline arrival + intensity value at hour 2.
  - Lunch Rush Intensity: This is how much more volume is expected for a lunch rush, it directly affects the baseline arrival rate by creating a peak of baseline arrival + intensity value at hour 5.

  - Randomness Intensity: This is how 'random' the arrival distribution will be, the larger the value the more the customers will derive from the expected arrival rates.
  - Morning Rush Duration: This is the time in minutes the morning rush will last for, the rush will start, grow to the peak arrival intensity, and simmer down once the rush ends.
  - Lunch Rush Duration: This is the time in minutes the lunch rush will last for, the rush will start, grow to the peak arrival intensity, and simmer down once the rush ends.

- Cost Weight Setup Parameters
  - Wait Time Penalty Weight: This is how much the cost function penalizes customer wait time, maxing the penalty after the customer waits 5 minutes, meaning the penalty lessens if the customer is served before 5 minutes.
  - Idle Time Penalty Weight: Penalty for the amount of time a barista is on the shift and not serving a customer.
  - Barista Count Penalty Weight: A labor cost, multiplied by 5 internally as we don't want to over schedule.
  - Dropped Customer Penalty Weight: Penalty weight for dropped customers, penalty is maxed when 10 customers get dropped.
  - Throughput Reward Weight: High throughput is good, max reward is achieved when 100 customers are served.
- Order Setup: This parameter is unique in the sense that it is inherently a table of orders. This could change the program entirely as far as revenue and costs go, Each drink can be assigned a price, time to make, a deviation of the time it takes to make, and a probability of it being ordered.

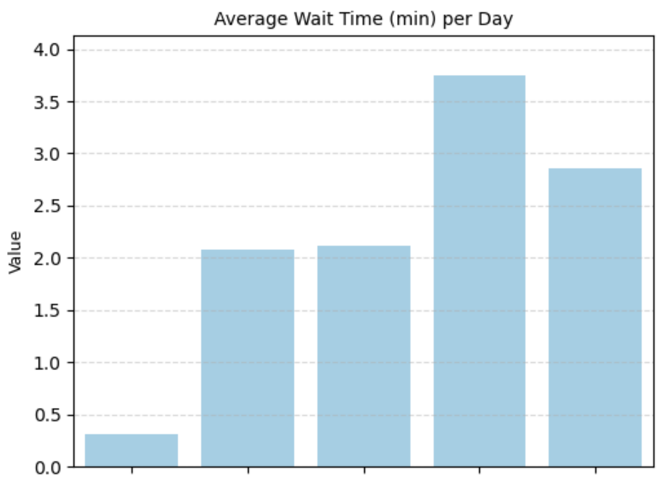
2.2. Which parameters matter most

As far as mentioning which parameters matter most, the baseline arrival rate is by far the most important parameter. This is the baseline parameter that determines how many customers will be coming through the cafe. This alone will affect the rest of the cafe operations ranging from revenue, throughput, dropped customers, and average customers in queue and allow analysis on how the scheduling manager and weights respond to an increase in volume.

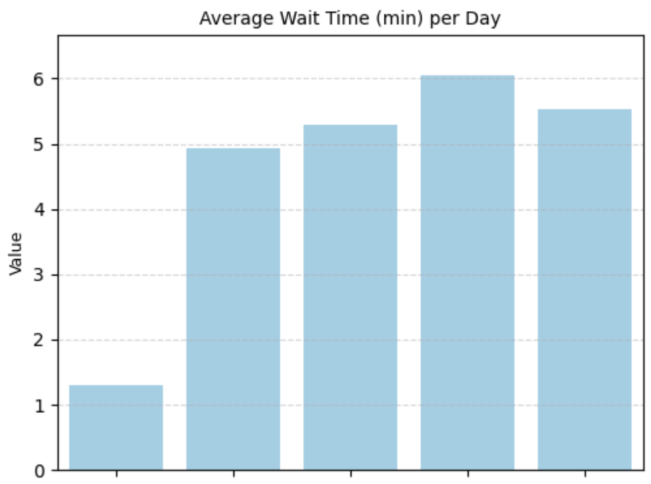
Another parameter that matters a lot is the Drink Probability parameter of the Orders table. The simulation can be easily manipulated with altering of the orders, higher priced drinks being more likely than any other drink could spike revenue, a high probability drink with a high service time could result in more customers dropping out of the queue or more baristas being scheduled.

There are some parameters that I added that do not provide much value to the simulation environment, the customer arrival process being as editable as it is makes the simulation seem overly complicated for the user, and the weight parameters do not seem to work as intended. Alteration of the weight values seemingly does not affect the amount of baristas being scheduled, meaning that a change to the scheduling manager is necessary for a functioning program.

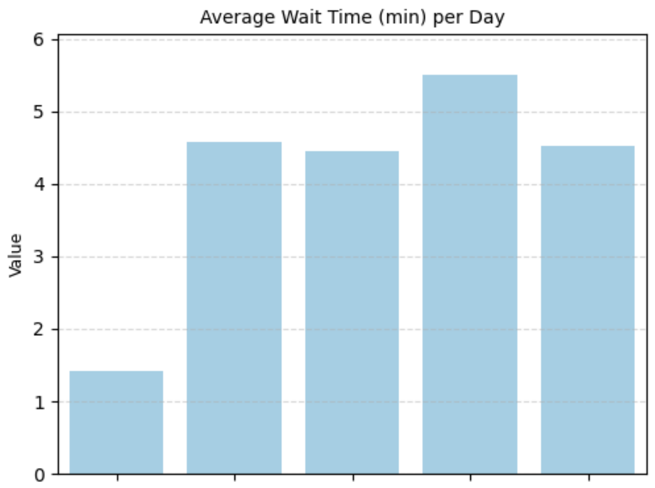
2.3. Tables/graphs showing relationships



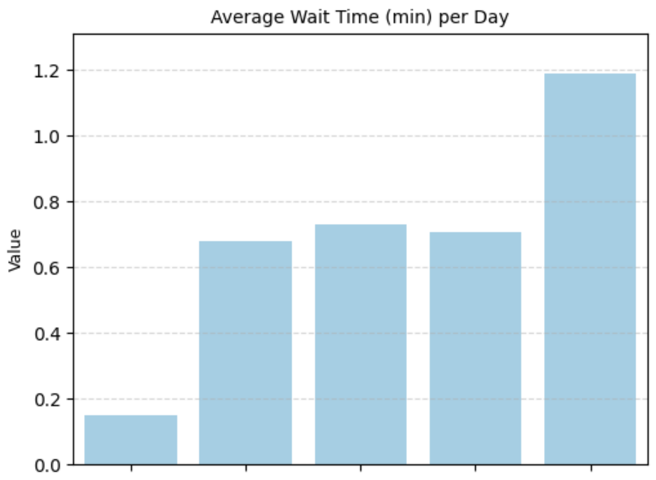
(a) Baseline wait time



(b) High Volume (baseline rate = 15)

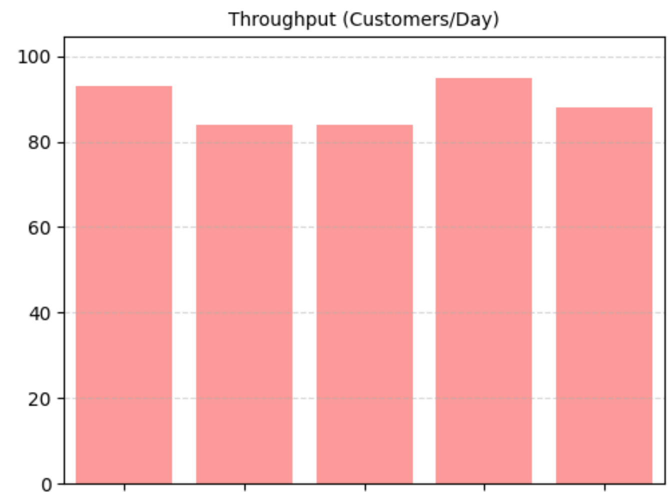


(c) High Intensity (Morning Intensity = 20, Lunch Intensity = 16)

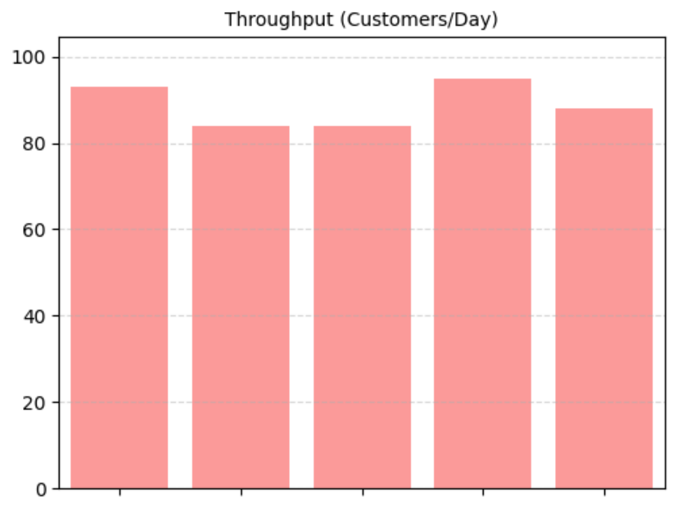


(d) No Rush (Morning/Lunch Intensity = 0)

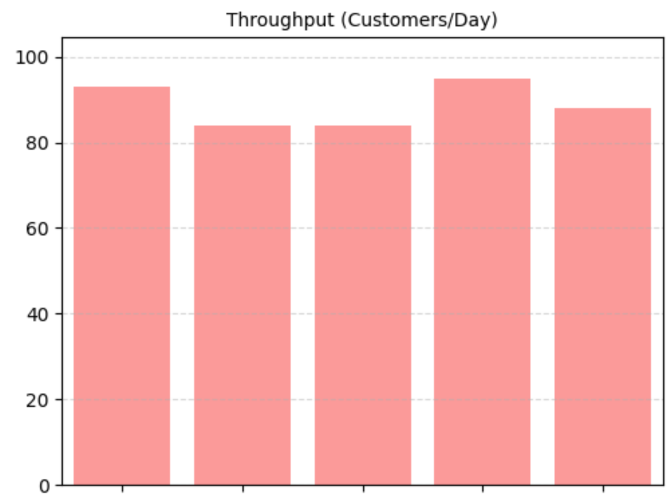
Figure 1: Average Wait Time across different arrival modifications



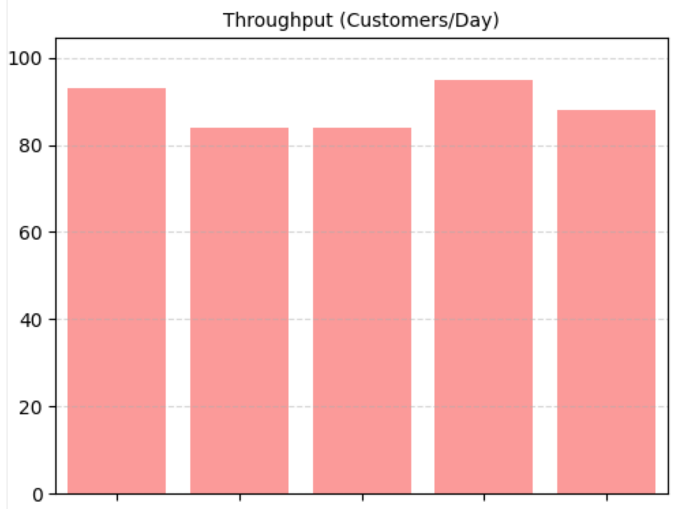
(a) Baseline Throughput



(b) No Barista Penalty (Barista Count Penalty = 0)



(c) Higher Barista Penalty (2)



(d) Minimal Barista Penalty (0.05)

Figure 2: Throughput across different barista penalties

3. Scenario Testing

3.1. Different conditions tested

This section covers scenarios that simulate what normal conditions look like as well as factors that could stress the system or cause any interesting edge cases.

Typical arrival rates vary depending on the nature of the area. A rural cafe would get much less traffic than a city cafe, and outside of city settings arrival peaks and rush durations will also vary. This simulation is crafted with the purpose of having flexible yet random arrivals, and on that notion normal conditions.

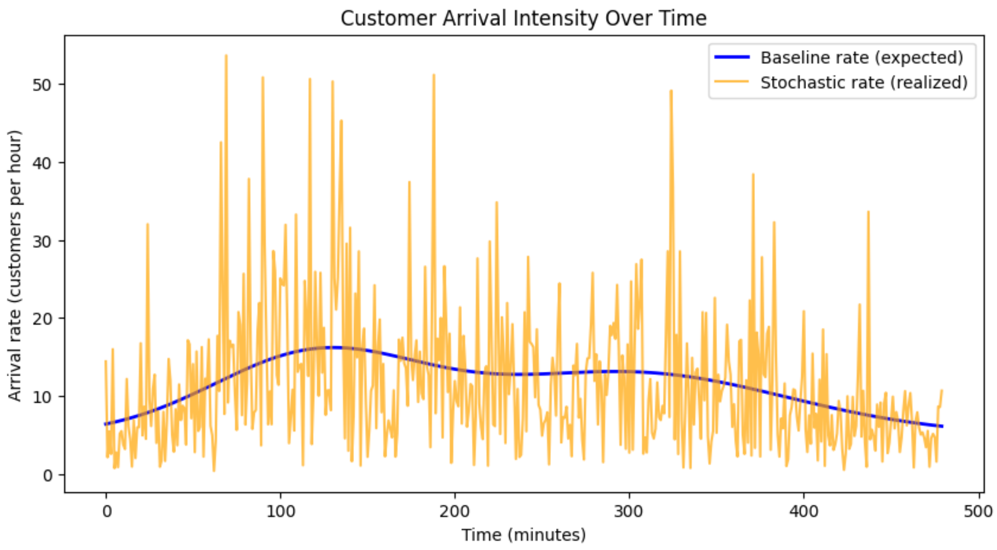


Figure 3: Simulated default arrivals

As can be seen from the arrival graph above, the higher peak but shorter duration morning rush overlaps with the longer lasting but less intense lunch rush, resulting in a simulated environment where the customer arrival rate fluctuates with predicted normal schedules.

Though, cafe operations outside of the arrival model vary. Though there is a cost-weight algorithm that determines the amount of baristas to schedule for the next day, the barista penalty is typically strong enough that the algorithm refuses to schedule more than one. This results in interesting cases where the schedule manager would rather drop over thirty customers as long as another barista would not have to be scheduled. It does not matter which parameters of the cost weight setup are changed, unless the value of the barisa count penalty weight negative, then it maxes out the baristas at 10.

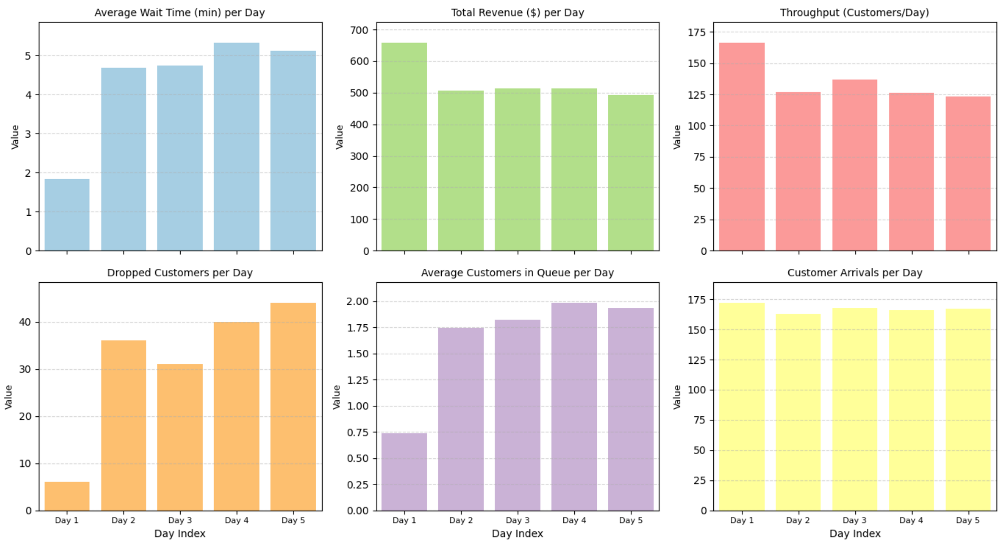


Figure 4: High throughput reward (10) and high baseline arrival (15)

Following this logic, the model does not react well to high stress. With plenty of customer arrivals, the schedule manager has no problem dropping many customers even at the cost of revenue. In accordance to the above graph, the cafe is missing out on an average of around $150 a day because it simply does not want to schedule another barista.

3.2. Results under each scenario

From M3, we have seen what is expected of the baseline scenario, which does result in a seemingly stable metrics across revenue, throughput, and wait times.

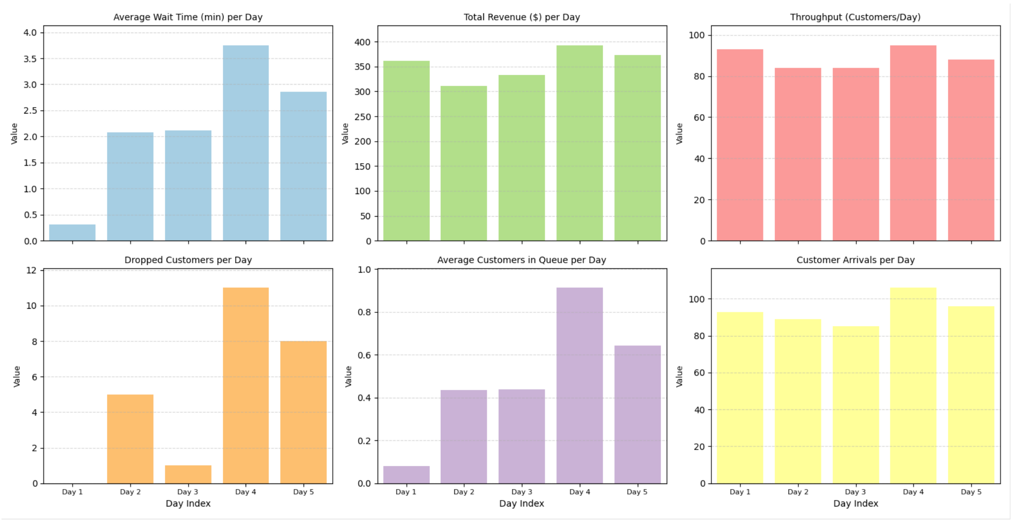


Figure 5: Metics returned from using default parameters

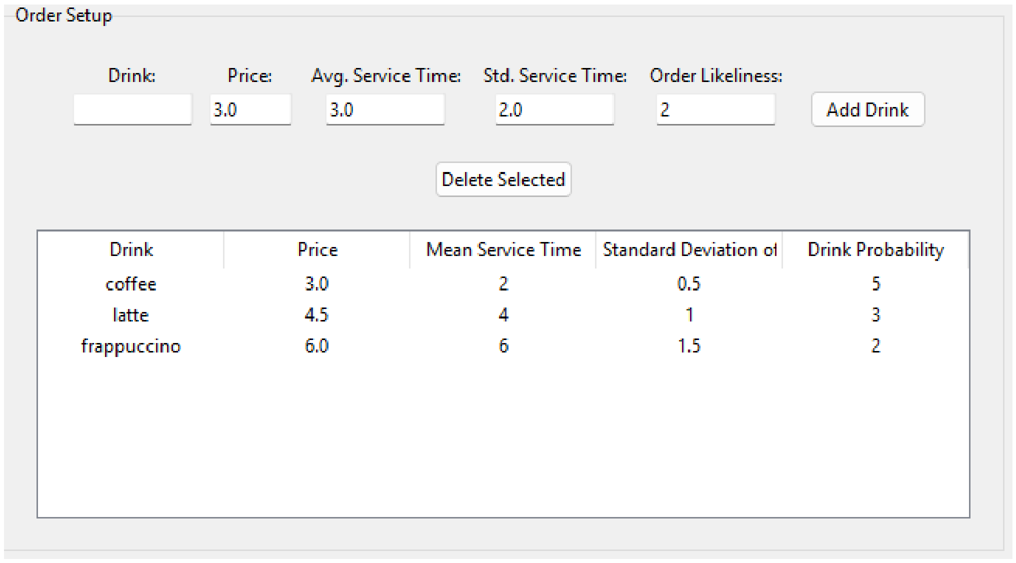
4. Validation

4.1. Evidence that model behaves reasonably

Face Validation: Queue and wait behavior match what is expected. As seen from the wait times across different arrival modifications, higher volume results in higher queue times.

We can see from Figure 1 of section 2.3 that queue and wait behavior acts as expected, meaning that the arrivals of the customers as well as the barista resource does work as intended. As the arrival rate grows, so does the wait time.

Parameter Validation: The default orders table used in all tests are logical; coffee is most ordered and does not take that much time, and is the cheapest to order. Lattes take slightly longer and also cost more. Frappuccinos take the longest and cost the most. The longer a drink takes the make, the more deviation occurs in the creation of the drink since there are more ingredients and steps that can take variable amounts of time. The arrival process is highly variable, depending on the setting of the cafe.


Order Setup

Drink:	Price:	Avg. Service Time:	Std. Service Time:	Order Likeliness:
	3.0	3.0	2.0	2

Add Drink

Delete Selected

Drink	Price	Mean Service Time	Standard Deviation of	Drink Probability
coffee	3.0	2	0.5	5
latte	4.5	4	1	3
frappuccino	6.0	6	1.5	2

Figure 6: Default Orders table

4.2. Comparison with expectations

The model does not meet my expectations for the simple reason that the scheduling manager does not want to schedule more than 1 barista.

Other than the fact that the model does not want to schedule more than 1 barista, the rest of the simulation does work as I intended. I am extremely happy with the customer arrival process, it is highly modifiable to meet real arrival expectations in accordance to real scenarios. The structuring of the orders table is executed well too, if I had real cafe statistics on how long drinks take to make and an actual menu, this could be extended upon to reflect a real menu.

4.3. Any limitations identified

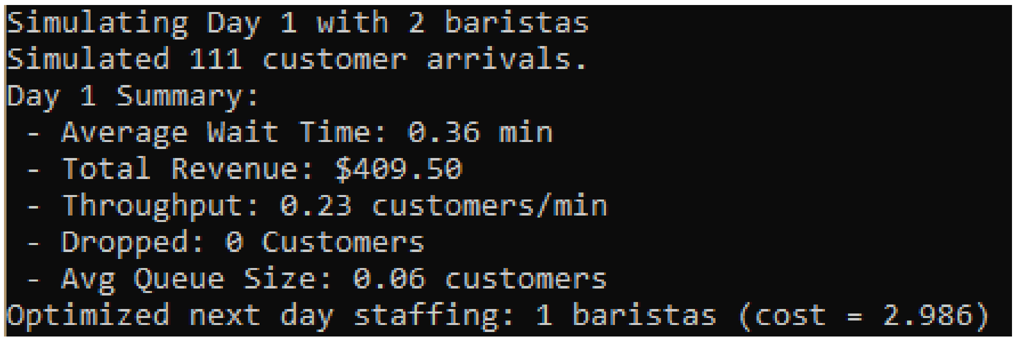
Limitations arise from the cost weight algorithm. If I were to redo the scheduling algorithm, I would have used a Monte Carlo simulation to predict and simulate possible futures rather than just using data from the day before for a simple algorithm to use a single metric to determine if another barista should be scheduled or not.

5. Statistical Summary

5.1. Key statistics from runs

Using the baseline parameters, I will calculate the mean, standard deviation, and range of all outputted metrics; average wait time, total revenue, throughput, dropped customers, average number of customers in queue, and customer arrivals per day. I will use 5 different runs, each one with a different seed. Each simulation run will consist of 6 days, the first day being omitted from calculations because of the initial scheduling of two baristas before the schedule manager decides to make a decision.

Metrics will be hard calculated. The graph is not the clearest when it comes to exact values, therefore I will use the console output in metric calculation. Said console outputs are placed into a list to calculate metrics from (use of TI-84 Plus CE).


```
Simulating Day 1 with 2 baristas
Simulated 111 customer arrivals.
Day 1 Summary:
 - Average Wait Time: 0.36 min
 - Total Revenue: $409.50
 - Throughput: 0.23 customers/min
 - Dropped: 0 Customers
 - Avg Queue Size: 0.06 customers
Optimized next day staffing: 1 baristas (cost = 2.986)
```
Figure 7: Sample command line output used to calculate statistics

Metrics Table				
Metric	Mean	Std Dev	Min	Max
Average Wait Time	2.938	0.438	2.19	3.68
Total Revenue	369.8	33.502	312	412.5
Throughput	0.169	0.014	0.14	0.19
Dropped Customers	9.267	2.987	5	15
Average Customers in Queue	0.496	0.109	0.3	0.66
Customer Arrivals	92	8.194	76	105

5.2. Sensitivity Testing

The purpose of this section is to cover how sensitive the baseline arrival parameter, randomness intensity parameter, and the throughput reward parameter are to the output metrics of the average wait time, average throughput, average arrivals, and average throughput.

The method I will use for each metric is through a single simulated run of 6 days, excluding calculations of the metrics obtained from the first day, leaving with a run sample size of 5 days. A second run will be generated using the same seed (1234567) with a 20% increase of the default parameter being tested (ex: default 10, test 12). Sensitivity will then be calculated by the percent change of the mean output divided by 20, as 20 is the percent change to the input.

Baseline arrival (Changed = 6)				
Metric	Base Mean	Changed Mean	Calculated % Change	Calculated Sensitivity
Average Wait Time	2.856	2.558	10.434%	0.522
Total Revenue	386.4	369.6	-4.35%	-0.217
Throughput	0.178	0.178	0	0
Dropped Customers	7.8	7.6	-2.564%	-0.128
Average Customers in Queue	0.512	0.440	-14.063%	-0.703
Customer Arrivals	96	93.4	-2.708%	-0.135

When picked apart, this makes almost no sense. The baseline arrival parameter went up, the actual arrivals went down. Even though the input parameter of the baseline arrivals was changed by 20%, the sensitivity of all of the recorded outputs was calculated as low sensitivity, meaning that the baseline arrival rate would have to change a significant amount if changes are to be observed in the arrival rates of customers. This goes against the face value validation that I have mentioned earlier, though when the parameter shows a significant increase in value, the amount of customer arrivals does end up higher.

Randomness Intensity (Changed = 2.4)				
Metric	Base Mean	Changed Mean	Calculated % Change	Calculated Sensitivity
Average Wait Time	2.856	4.058	42.087%	2.104
Total Revenue	386.4	450.3	16.537%	0.827
Throughput	0.178	0.194	8.989%	0.449
Dropped Customers	7.8	19.2	146.154%	7.308
Average Customers in Queue	0.512	0.852	66.406%	3.320
Customer Arrivals	96	112.6	17.292%	0.865

Judging from the statistics gathered from the randomness intensity table, the randomness intensity parameter is highly unpredictable. Even though sensitivity towards customer arrivals was not high, wait time, drop count, and average number of customers in queue were highly sensitive to randomness. From this information, I could derive that more sharper groupings were calculated. Below is a chart of the customer arrival intensity that can be compared with the baseline figure 3 intensity graph. Notice the range of the y-axis has significantly grown, when it would peak at 50 we now see peaks at 70.

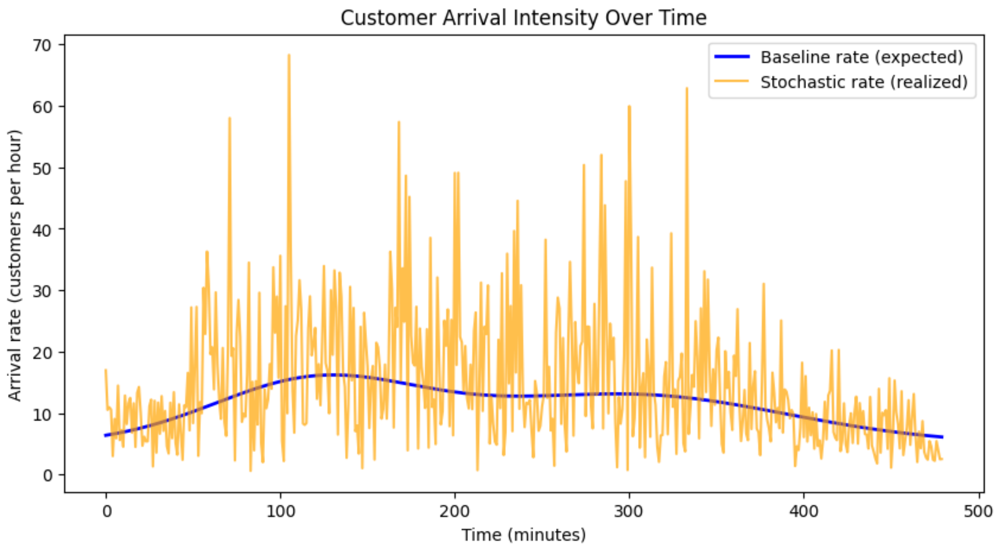


Figure 8: Metics returned from using default parameters

Throughput Reward (Changed = 2.4)				
Metric	Base Mean	Changed Mean	Calculated % Change	Calculated Sensitivity
Average Wait Time	2.856	2.856	0	0
Total Revenue	386.4	386.4	0	0
Throughput	0.178	0.178	0	0
Dropped Customers	7.8	7.8	0	0
Average Customers in Queue	0.512	0.512	0	0
Customer Arrivals	96	96	0	0

As I predicted, changing the throughput reward weight did not cause more than 1 barista to be scheduled, therefore the output of the metrics would not change. Even pushing the throughput reward up to 50 did not bring a change to the outputs. I fear that it is a useless parameter that has no sensitivity to the simulation therefore also providing no use as a parameter to the simulation.

6. Conclusions

6.1. Main findings

The simulation demonstrates that the arrival model and order-table mechanics behave in an expected and interpretable manner: increasing effective arrival intensity increases wait times and queue occupancy, and more time-consuming drinks correlate with higher service-time variance and throughput pressure. The summary statistics indicate the model produces stable, measurable outputs suitable for comparative experiments.

However, several parameters have outsized or unexpected effects on outcomes: the Randomness Intensity parameter yields large sensitivity (substantial increases in wait time, dropped customers, and queue size when randomness is increased), while the Throughput Reward and cost-weight parameters meant to control scheduling produce little to no change in staffing because the scheduling manager consistently chooses one barista under current logic. Finally, there are interesting edge cases considering the sensitivity analysis (for example, increases in baseline arrival rate not always producing consistent increases in recorded arrivals) that point to either sampling variability under small sample sizes or implementation bugs.

6.2. Model strengths and weaknesses

Strengths:

- Flexible arrival model: the doubly-stochastic arrival process with configurable baseline rate, rush intensities and durations, and randomness intensity creates realistic and diverse day profiles suitable for stress tests and scenario experiments.
- Intuitive order table: the order table structure (price, mean service time, time deviation, order probability) maps cleanly to revenue and service-time behavior, making it straightforward to model menu changes and study their operational impact.
- Clear metrics and visualizations: the simulation records, prints, and visualizes key KPIs (average wait time, throughput, dropped customers, revenue, arrivals, average queue size), enabling objective comparison across scenarios.

Weaknesses

- Scheduling optimizer breakdown: the cost-weight based schedule manager behaves incorrectly — the barista count penalty dominates decisions so strongly that the algorithm repeatedly refuses to schedule more than one barista even under heavy load, causing excessive dropped customers and lost revenue. This undermines the central goal of studying scheduling tradeoffs.
- Lack of predictive staffing: the current scheduling strategy uses only the previous day's data and a single scalar objective, making it reactive rather than predictive. This leads to poor performance under variable arrivals.

6.3. Recommendations

Overall, the simulation provides a strong foundation for studying the cafe's daily operations, but the results highlight several areas where refinement would meaningfully improve accuracy and decision-making. The most important step would be to revisit the scheduling cost function, which currently over-penalizes staffing and causes the optimizer to choose one barista regardless of any gathered metric. Adjusting the weight scaling and verifying each term's contribution would allow the scheduler to respond more realistically to changes in demand.

Additionally, some sensitivity tests produced inconsistent responses to changes in arrival parameters, suggesting that parts of the arrival model or its parameter plumbing should be validated. Ensuring that baseline rate, rush intensities, and randomness levels propagate correctly through the arrival generator would increase confidence in the model's behavior.

In the longer term, the model would benefit from a more predictive scheduling approach. Rather than relying solely on the previous day's metrics, incorporating forward simulation or Monte Carlo based evaluation would help identify staffing levels that minimize expected cost across multiple possible demand scenarios. Refinements such as including net profit (revenue minus labor cost) and testing the simulation against queuing-theory baselines would strengthen the model's validity.